ILKOM Jurnal Ilmiah
Vol. 12 No. 3, Desember 2020, pp.176-182
E-ISSN 2548-7779

Damerau Levenshtain Distance dengan Metode Empiris untuk Koreksi Ejaan Bahasa Indonesia

Aji Prasetya Wibawa[a,1,*], Pundhi Yuliawati [a,2], Puji Santoso [a,3], Ridwan Shalahuddin [a,4] dan I Made Wirawan [a,5]

[a]*Electrical Engineering Department, Universitas Negeri Malang, Malang, Indonesia*
[1]*aji.prasetya.ft@um.ac.id;* [2]*pe.yhulia@gmail.com;* [3]*pujiosotnas@gmail.com;* [4]*ridhwan102@gmail.com;*
[5]*made.wirawan.ft@um.ac.id*
**corresponding author*

INFORMASI ARTIKEL

Dikirim : 01 Juli 2020
Diulas : 23 Juli 2020
Direvisi : 23 Agustus 2020
Diterbitkan : 28 Desember 2020

Kata Kunci:
Damerau Levenshtain Distance
Metode Empiris
Ejaan
Koreksi
Bahasa Indonesia

ABSTRAK

Damerau Levenshtein Distance (DLD) adalah algoritma untuk koreksi kesalahan penulisan. Kesalahan terjadi karena penyisipan, penghapusan, pertukaran, dan penggantian alfabet dalam sebuah kata. Ini mungkin terjadi karena hilangnya spasi di antara dua kata. DLD tidak dapat mengatasi masalah kehilangan spasi. Karenanya, makalah ini bertujuan untuk menggabungkan DLD dengan Metode Empiris untuk memperbaiki kesalahan ini. Alhasil, algoritma kombinasi dapat mengungguli DLD asli dalam memeriksa kesalahan ejaan Teks Bahasa Indonesia dengan akurasi 97%.

Keywords:
Damerau Levenshtain Distance
Empirical Method
Spelling
Correction
Indonesian

ABSTRACT

Damerau Levenshtein Distance (DLD) is an algorithm for writing-error correction. The errors happened due to insertions, deletion, exchange, and substitution of alphabet within a word. It may occur because of the loss of space between two words. DLD alone is unable to overcome the loss psace problems. Thus, this paper aims to combine DLD with Empirical Method to fix this specific error. As a result, the combination algorithm may outperform the original DLD in checking the spelling errors of Indonesian Text with an accuracy of 97%.

I. Pendahuluan

Damerau Levenshtain Distance (DLD) merupakan salah satu pengembangan dari algoritma *Levenshtein Distance* [1], [2]. DLD dapat mengoreksi kesalahan penulisan seperti hilangnya karakter huruf, kelebihan karakkter huruf, dan kesalahan urutan huruf [3], [4]. Perbaikan dilakukan dengan cara merubah sebuah string menjadi string yang lain. Untuk melakukan hal tersebut DLD akan menjalankan operasi *insertion*, *deletion*, *substitution*, *dan transposition* [5]. Metode praktis ini telah banyak digunakan untuk memperbaiki kesalahan penulisan [6], memperbaiki *text alignment* [7], *information retrieval* [8], dan menemukan kemiripan (*similarity*) dalam teks [9].

Pengorekasian ejaan berbahasa menggunakan DLD mendapatkan hasil yang cukup akurat, yaitu 73% [10]. Upaya optimasi juga dilakukan dengan mempercepat waktu komputasi tanpa mengurangi akurasi algoritma. Baris dan kolom pertama pada matriks DLD dihapus untuk mempercepat proses perhitungan jarak yang berulang [6]. Keduanya mampu mengoreksi ejaan bahasa Indonesia pada kata-kata sederhana. Di sisi lain, DLD tidak bisa begitu saja mengoreksi kata-kata yang ditulis berdempet (tanpa spasi). Kata-kata ini tidak ditemukan keberadaannya dalam database yang dibuat berdasarkan KBBI. Akibatnya, saran koreksi yang dihasilkan tidak tepat,

Salah satu metode yang mungkin diterapkan untuk mengatasi masalah ini adalah metode empiris. Metode ini mampu dalam mengoreksi kata yang berdempetan akibat kesalahan penulisan yaitu kurangnya spasi [11]. Artikel ini bertujuan untuk meningkatkan kinerja DLD dalam mengenali kesalahaan penulisan kata yang tidak

http://dx.doi.org/10.33096/ilkom.v12i3.600.176-182

seharusnya berdempetan. Metode ini akan dibandingkan dengan DLD [10] untuk mengetahui algoritma yang lebih efisien dalam mengoreksi ejaan bahasa Indonesia.

II. Metode

Penelitian ini terdiri dari lima tahapan penelitian. Tahap pertama dimulai dengan pengumpulan data set sebagai input proses pengoreksian. Pada tahap kedua dilakukan pre-processing pada dataset untuk medapatkan data yang sesuai dengan kebutuhan sistem pengoreksi ejaan. Pada tahap ketiga dilakukan pengoreksiaan ejaan dengan menggunakan algoritma DLD. Tahap yang keempat menjalankan metode empiris untuk memecah kata. Tahap yang kelima yaitu menampilkan saran kata yang benar sesuai dengan kamus KBBI. Kemudian dilakukan perhitungan hasil akurasi. Gambar 1 menampilkan algoritma gabungan DLD dan metode empiris.

Penelitian ini menggunakan Dataset berasal dari dua dongeng yang berasal dari ceritadongengrakyat.com. Dataset yang digunakan sebanyak 1266 kata yang memiliki kesalahan penulisaan ejaan kata sehingga perlu dilakukan perbaikan. Kata yang dianggap salah adalah kata-kata tertulis dalam dongeng namun tidak terdaftar pada KBBI (Kamus Besar Bahasa Indonesia). Kesalahan ejaan disebabkan oleh beberapa hal, seperti ketidaktahuan dalam penulisan, kesalahan pada saat penulisan atau pengetikan dan kesalahan pada mesin pada saat penyimpanan [12], [13].

Secara rinci contoh kesalahan yang sering terjadi pada saat pengetikan atau penulisan sebagai berikut:

1. Kurang pengetikan huruf pertama di awal kata; Contoh: Kata ‘Gurita’ menjadi ‘Urita’.
2. Kurang pengetikan huruf pertama di tengah kata; Contoh: Kata ‘Wayang’ menjadi ‘Wayng’.
3. Kurang pengetikan huruf pertama di akhir kata; Contoh: Kata ‘Wayang’ menjadi ‘Wayan’.
4. Kelebihan pengetikan huruf pertama di awal kata; Contoh: Kata ‘Wayang’ menjadi ‘Wwayang’.
5. Kelebihan pengetikan huruf pertama di tengah kata; Contoh: Kata ‘Wayang’ menjadi ‘Wayyang’.
6. Kelebihan pengetikan huruf pertama di akhir kata; Contoh: Kata ‘Wayang’ menjadi ‘Wayangg’.
7. Posisi huruf tertukar; Contoh: Kata ‘Wayang’ menjadi ‘Waynag’.
8. Kurang spasi pada kata yang berdempetan; Contoh: Kata ‘makan nasi’ menjadi ‘makannasi’

Tahapan berikutnya, *pre-pocessing*, menghilangkan tanda baca yang terdapat pada teks cerita dongeng. Selain itu, perlu memasukkan kata-kata berimbuhan kedalam kamus basis data. Kamus yang dipakai hanya terdapat kata dasar sehingga ketika memproses kata berimbuhan terjadi eror. Untuk mengatasi hal tersebut perlu dilakukan input kata berimbuhan kedalam kamus basis data. Contoh kata-kata berimbuhkan yang perlu DIMASUKKAN KEDALAM KAMUS BASIS DATA ADALAH MENJAGA, BERDUA, KEBAHAGIAAN, PEMALAS, TERPERANJAT, menjelaskan, menutup, memperkenalkan, berdiri, dan membuat. Kata-kata ini terdiridari kata dasar yang memiliki imbuhan berupa awalan, akhiran, sisipan serta kombinasi ketiganya.

```
#Algoritma kombinasi DLD Empirik
BEGIN
    Kerjakan preprocessing
    Kerjakan DLD
    IF jarak <=2
        Tampilkan saran Kata
    ELSE
        Kerjakan metode empiris //memecah kata menjadi dua
            IF kata1 ditemukan
                Tampilkan kata1 dan saran kata2
            IF kata2 ditemukan
                Tampilkan kata2 dan saran kata1
            IF kata1 dan kata2 ditemukan
                Tampilkan kata1 dan kata2
            ELSE Tampilkan “Kata tidak ditemukan”
END
```

Gambar 1. Algoritma gabungan DLD dan metode empiris

Langkah-langkah algoritma DLD[14], [15] pada Gambar1 dapat dijabarkan sebagai berikut.

1. Inisialisasikan n sebagai panjang karakter dari s dan m sebagai panjang karakter dari t. Jika n = 0 atau m = 0, maka kembalikan nilai (return value) berupa jarak_edit = max(n, m); lalu lompat ke langkah 7.
2. Buat sebuah matriks d sebanyak m + 1 baris dan n + 1 kolom.
3. Isi baris pertama dengan 0..n dan isi kolom pertama dengan 0..m.
4. Periksa setiap karakter dari s terhadap t

   Jika s[i] = t[j] maka cost = 0.

   Jika s[i] ≠ t[j] maka cost = 1.
5. Isikan nilai dari setiap sel d[i, j] baris per baris dengan:

   d[i, j] = min(x, y, z)

   keterangan :

   d[i, j] : sel yang merupakan pertemuan kolom j dengan baris i pada matriks d.

   x : nilai yang terdapat di sel atas dari posisi sel sekarang ditambah dengan 1 (satu) atau dapat dirumuskan : x = d[i - 1, j]+1

   y : nilai yang terdapat di sel sebelah kiri dari posisi sel sekarang ditambah 1 (satu) atau dapat dirumuskan : y = d[i, j - 1] + 1

   z : nilai yang terdapat di sel sebelah atas dari sebelah kiri sel sekarang (arah barat laut) ditambah nilai cost dan dapat dirumuskan : z = d[i – 1, j – 1] + cost

   Jika i > 1 dan j > 1 dan s[i] = t[j - 1] dan s[i - 1] = t[j] yang mana artinya setelah kedua kata dibandingkan terdapat karakter yang dapat ditransposisikan, maka isi nilai sel d[i, j] dengan rumusan berikut:

   d[i, j] = min(d[i, j], d[i - 2, j - 2] + cost)
6. Setelah langkah iterasi selesai, maka jarak edit akan ditemukan pada sel d[n, m] yaitu sel pada pojok kanan baris terakhir.
7. Proses DLD selesai.

Gambar 2 merepresentasikan matriks DLD mxn. Pada contoh ini, m dan n = 6. Pada contoh, kata kunci dan kata yang dikoreksi memiliki panjang karakter yang sama. Kesalahan ejaan hanya terdapat pada dua huruf yang posisinya terbalik. Pemeriksaan karakter dilakukan mulai s terhadap t. Isikan nilai dari setiap sel *d[i.j]* baris perbaris. Langkah ini akan terus berulang sampai semua matriks terisi (Gambar 2). Dari contoh kata “KAMPUS” dan “KAMPSU” hanya memiliki satu jarak perbedaan nilai jarak. Pengoreksian untuk contoh ini hanya memperlukan satu operasi yaitu penukaran dua huruf yang berdekatan unuk merubah KAMPSU menjadi KAMPUS.

S/T		K	A	M	P	U	S
	0	1	2	3	4	5	6
K	1	0	1	2	3	4	5
A	2	1	0	1	2	3	4
M	3	2	1	0	1	2	3
P	4	3	2	1	0	1	2
S	5	4	3	2	1	1	1
U	6	5	4	3	2	1	1

Gambar 2. Contoh matriks DLD 6X6

Pengoreksian kata berdempetan menggunkan Metode Empiris bekerja dengan cara memisahkan string menjadi beberapa kemungkinan kata kemudian dicocokan ke dalam basis data. Misalnya penulisan kata "makannasi", jika dicari dalam kamus Bahasa Indonesia tidak memiliki arti. Cara kerja metode ini adalah mecoba semua kemungkinan yang ada. Tabel I dapat dilihat bahwa kata "makan" dan "nasi" memiliki arti dalam Bahasa Indonesia dan terdaftar dalam basis data. Saran perbaikan yang diberikan yaitu "makan nasi". String dipecah dengan menyisipkan spasi pada setiap kemungkinan sampai menghasilkan saran perbaikan yang sesuai.

Tabel 1. Pembentukan dan pencarian kata

Langkah	Kata1	Kata2	Keterangan
1	m	akannasi	Tidak terdapat dalam kamus
2	ma	kannasi	Tidak terdapat dalam kamus
3	mak	annasi	Tidak terdapat dalam kamus
4	maka	nnasi	Tidak terdapat dalam kamus
5	makan	nasi	Terdapat dalam kamus

Saran kata ini berupa tampilan perbaikan dari pengoreksian kata yang salah. Hasil dari saran kata yang ditampilkan yaitu kata yang terdapat dalam kamus basis data yang paling mirip dengan kata yang salah. Saran kata ini sebagai hasil dari perbaikan akibat penulisan kata yang salah. Evaluasi dilakukan dengan menghitung akurasi [10].

III. Hasil dan Pembahasan

Hasil pengujian dengan kesalahan penulisan sejumlah 100 ditampilkan pada Tabel 2. Hasil pengujian diperoleh bahwa DLD tidak dapat mengoreksi kesalahan kata yang berdempetan seperti pada Tabel 3. Setelah dilakukan penambahan dengan Metode Empiris hasil pengujian meningkat menjadi 93%. Dua kata yang berdempetan karena hilang spasi dapat dikoreksi. Tabel 4 menunjukkan contoh hasil koreksi kombinasi DLD dengan metode empiiris.

Tabel 2. Perbandingan akurasi algoritma

Metode	Kesalahan Ejaan	Kesalahan dapat dikoreksi	Kesalahan tidak dapat dikoreksi	Akurasi
DLD	100	73	27	73%
DLD+Empiris	100	97	3	97%

Berdasarkan hasil dari Tabel 2, dari 100 kesalahan yang terjadi diketahui bahwa 97 kesalahan dapat dikoreksi dan 3 kesalahan tidak dapat dikoreksi. Tiga kesalahan tersebut tidak dapat dikoreksi karena metode empiris tidak mampu mengoreksi kata yang salah satunya terjadi kesalahan penulisan. Contoh hasilnya ditunjukkan pada Tabel 4.

Tabel 3. Contoh hasil pengujian DLD

Kesalahan Kata	Saran Perbaikan dengan DLD
putriraja	putriraja
adalahtitisan	adalahtitisan
seoranggadis	seoranggadis
sebuahgunung	sebuahgunung
melakukankesalahan	melakukankesalahan
yangsangat	yangsangat
denganmarah	denganmarah
sangtkesal	sangtkesal
bekaslukaa	bekaslukaa
makhlukghaib	makhlukghaib

DLD mampu mengoreksi berbagai macam kesalahan penulisan ejaan. Kelemahan dari metode tersebut, tidak dapat mengoreksi hilangnya spasi antara dua kata. Sehingga penelitian ini ditambah dengan Metode Empiris.

Tabel 4. Contoh hasil pengujian metode empiris

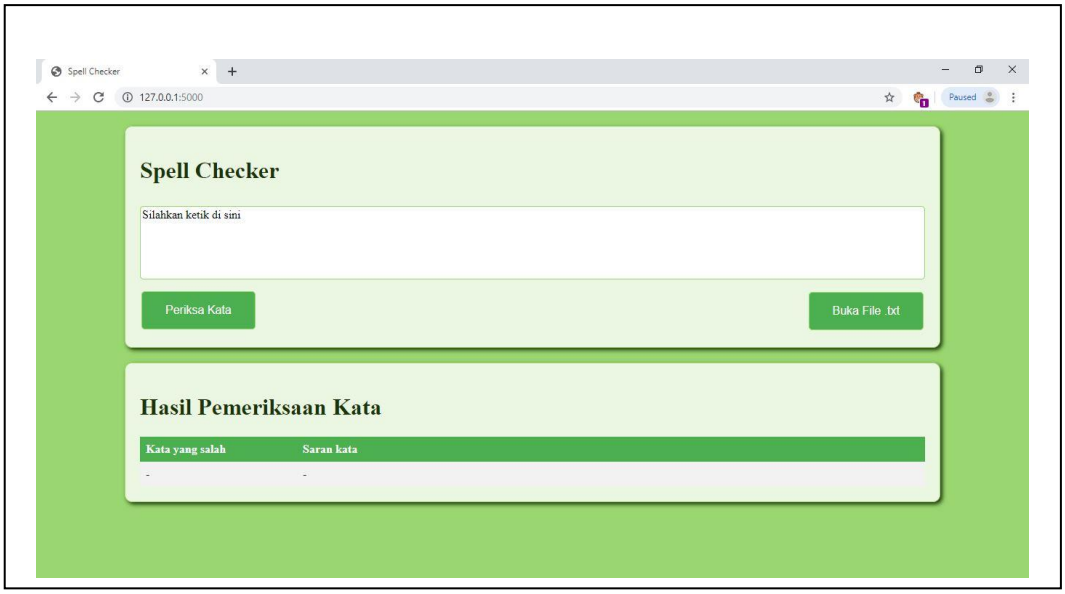
Kesalahan Kata	Saran Perbaikan dengan DLD	Saran Perbaikan dengan Empiris
putriraja	putriraja	putri raja
adalahtitisan	adalahtitisan	adalah titisan
seoranggadis	seoranggadis	seorang gadis
sebuahgunung	sebuahgunung	sebuah gunung
melakukankesalahan	melakukankesalahan	melakukan kesalahan
yangsangat	yangsangat	yang sangat
denganmarah	denganmarah	dengan marah
sangtkesal	sangtkesal	sangtkesal
bekaslukaa	bekaslukaa	bekaslukaa
makhlukghaib	makhlukghaib	makhlukghaib

Tabel 4 menunjukan hasil pengujian dengan metode empiris. Metode empiris tidak mampu mengoreksi kata apabila kata yang berdempetan salah satunya terdapat kesalahan penulisan. Misalnya pada kata "makhlukghaib" saran perbaikan yang ditampilkan yaitu "makhlukghaib". Hal tersebut karena penulisan kata "ghaib" terdapat kesalahan yang seharusnya ditulis "gaib". Untuk mengatasi hal tersebut perlu dilakukan penelitian selanjutnya dengan cara melakukan modifikasi seperti pada Gambar 3. Pada Gambar 3 DLD dieksekusi kembali jika terdapat kata hasil pecahan yang tidak terdapat dalam database. Hal ini juga memungkinkan pembetulan lebih dari dua kata yang berimpit karena kehilangan spasi.

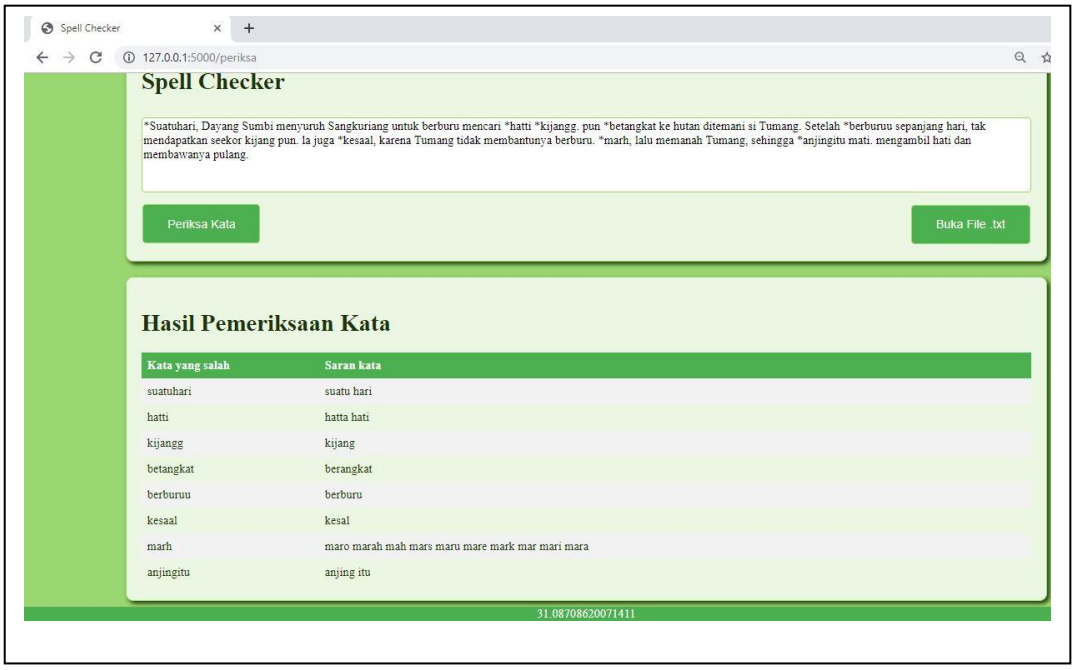
```
#ModifikasiAlgoritma kombinasi DLD Empirik
BEGIN
    Kerjakan preprocessing
    Kerjakan DLD
    IF jarak <=2
        Tampilkan saran Kata
    ELSE
        Kerjakan metode empiris //memecah kata menjadi dua
            IF kata1 ditemukan
                Lakukan DLD pada kata2
                Tampilkan kata1 dan saran kata2
            IF kata2 ditemukan
                Lakukan DLD pada kata1
                Tampilkan kata2 dan saran kata1
            IF kata1 dan kata2 ditemukan
                Tampilkan kata1 dan kata2
            ELSE Tampilkan "Kata tidak ditemukan"
END
```

Gambar 3. Modifikasi gabungan DLD dan metode empiris

Gambar 4 menampilkan tampilan system pengoreksi ejaan berbasis web. Bagian atas digunakan untuk menuliskan teks yang akan dikoreksi secara manual (pengetikan). Terdapat tombol periksa untuk memproses pengoreksian teks dan tombol buka file.txt untuk menginputkan teks yang akan dikoreksi. Pada bagian bawah menampilkan hasil proses pengoreksian kesalahan ejaan kata. Kata yang salah pada kolom sebelah kiri dan saran kata pada kolom sebelah kanan. Hasil pemrosesannya ditampilkan pada Gambar 5.



Gambar 4. Tampilan sistem koreksi ejaan



Gambar 5. Modifikasi gabungan DLD dan metode empiris

IV. Kesimpulan

Modifikasi algoritma *Damerau Levenshtein Distance* dan metode empiris untuk mengoreksi ejaan bahasa Indonesia pada cerita dongeng menghasilkan saran perbaikan yang baik dengan akurasi 97%. Namun masih terdapat beberapa kelemahan sehingga perlu dilakukan pengembangan penelitian selanjutnya. Sistem ini hanya menghasilkan satu saran kata satu untuk setiap kesalahan. Pengoreksian spasi pada kata yang berdempetan perlu dilakukan penelitian dengan metode lain untuk mengatasi kesalahan pada salah satu katanya sehingga menghasilkan perbaikan kata yang akurat. Pendekatan heuristic adalah salah satu cara yang dapat menyempurnakan kinerja sistem ini.

Ucapan Terima Kasih

Penelitian ini dibawah naungan grup penelitian, Knowledge Engineering and Data Science (KEDS) yang berada di Program Studi Teknik Informatika, Jurusan Teknik Elektro, Universitas Negeri Malang (UM). Terimakasih kami ucapkan kepada Profesor Suyono dari jurusan bahasa Indonesia UM yang telah meluangkan waktu untuk berdiskusi dan memvalidasi hasil penelitian ini.

Daftar Pustaka

[1] A. I. Fahma, I. Cholissodin, and R. S. Perdana, "Identifikasi Kesalahan Penulisan Kata (Typographical Error) pada Dokumen Berbahasa Indonesia Menggunakan Metode N-gram dan Leven shtein Distance," *J. Pengemb. Teknol. Inf. dan Ilmu Kompute*, vol. 2, no. 1, pp. 53–62, 2018.

[2] A. Pahdi, "Koreksi Ejaan Istilah Komputer Berbasis Kombinasi Algoritma Damerau- Levenshtein dan Algoritma Soundex," *Sentra Penelit. Eng. dan Edukasi*, vol. 8, no. 2, pp. 1–8, 2016.

[3] G. V. Bard, "Spelling-error tolerant, order-independent pass-phrases via the Damerau-Levenshtein string-edit distance metric," in *the fifth Australasian symposium on ACSW frontiers*, 2007, pp. 117–124.

[4] A. S. Lhoussain and Y. O. U. S. F. I. Hicham, G.U.E.D.D.A.H. Abdellah, "Adaptating the levenshtein distance to contextual spelling correction," *Int. J. Comput. Sci. Appl.*, vol. 12, no. 12, pp. 127–133, 2015.

[5] H. Hyyrö, "A bit-vector algorithm for computing Levenshtein and Damerau edit distances," *Nord. J. Comput*, vol. 10, no. 1, pp. 29–39, 2003.

[6] P. Santoso, P. Yuliawati, R. Shalahuddin, and I. A. E. Zaeni, "Penghapusan kolom dan baris pertama pada matriks distance untuk optimasi spell checker damerau-levenshtein distance," *Sains, Apl. Komputasi dan Teknol. Inf.*, vol. 2, no. 2, pp. 57–63, 2020.

[7] A. Kutuzov, "Improving English-Russian sentence alignment through POS tagging and Damerau-Levenshtein distance," in *Proceedings of the 4th Biennial International Work*, 2013, pp. 63–68.

[8] G. Veena and G. Jalaja, "No TitleLevenshtein Distance based Information Retrieval.," *Int. J. Sci. Eng. Res.*, vol. 6, no. 5, 2015.

[9] S. Y. Yuliani, S. Sahib, M. F. Abdollah, Y. S. Wijaya, and N. H. M. Yusoff, "Hoax news validation using similarity algorithm," *J. Phys. Conf. Ser.*, vol. 1524, no. 1, p. 012035, 2020.

[10] P. Santoso, P. Yuliawati, R. Shalahuddin, and A. P. Wibawa, "Damerau Levenshtein Distance for Indonesian Spelling Correction," *J. Inform.*, vol. 13, no. 2, p. 11, 2019.

[11] N. M. M. Adriyani, I. W. Santiyasa, and A. Muliantara, "Implementasi Algoritma Levenshtein Distance dan Metode Empiris untuk Menampilkan Saran Perbaikan Kesalahan Pengetikan Dokumen Berbahasa Indonesia," *JELIKU (Jurnal Elektron. Ilmu Komput. Udayana)*, vol. 1, no. 1, 2012.

[12] N. Gupta and P. Mathur, "Spell checking techniques in NLP: a survey," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 2, no. 12, pp. 217–221, 2012.

[13] V. V. Bhaire, A. A. Jadhav, and P. G. Pasthe, Pradnya A. Magdum, "Spell checker," *Int. J. Sci. Res. Publ.*, vol. 5, no. 4, pp. 1–3, 2015.

[14] F. J. Damerau, "A technique for computer detection and correction of spelling errors," *Commun. ACM*, vol. 7, no. 3, pp. 171–176, 1964.

[15] R. Gabrys, E. Yaakobi, and O. Milenkovic, "Codes in the Damerau distance for DNA storage," in *2016 IEEE International Symposium on Information Theory (ISIT)*, 2016, pp. 2644–2648.